Enhanced VoiceXML

José Rouillard, Philippe Truillet

Enhanced VoiceXML

José Rouillard & Philippe Truillet

Laboratoire Trigone - CUEEP - Bat B6
Université des Sciences et Technologies de Lille
59655 Villeneuve d'Ascq cedex - France
Email: jose.rouillard@univ-lille1.fr
Telephone: (33) 03.20.43.32.70
Fax: (33) 03.20.43.32.79

Laboratoire IRIT
Université Paul Sabatier
118, Route de Narbonne – Toulouse Cedex 4
Email: Philippe.Truillet@irit.fr
Telephone: (33) 05.61.55.74.08
Fax: (33) 05.61.55.62.58

Abstract

This article presents a reflection on an extension of VoiceXML language. VoiceXML applications are described by context-free grammars. Then, recognized vocabulary is limited. The idea is to overlap this limitation of VoiceXML by using asynchronously capabilities of speech-dictation systems. A prototype of extended VoiceXML system is presented. According to us, that opens many tracks of development on vocal systems.

1 Introduction

Vocal technologies and interfaces are an important part of Human-Computer Interaction (HCI) area. Using natural language within the interaction is supposed to facilitate human-machine exchanges.

Domains such as E-health, E-learning, E-trade, M-trade or E-administration are many sectors for which simple and efficient vocal interactions are waited. These factors pushed the W3C to work on this aspect and to publish a recommendation concerning a vocal interaction language, based on XML, which allows describing and managing vocal interactions on the Internet network. Indeed, VoiceXML is a programming language, designed for human-computer audio dialogs that feature synthesized speech, digitized audio, recognition of spoken and DTMF key input, recording of spoken input, telephony, and mixed initiative conversations. Its major goal is to bring the advantages of web-based development and content delivery to interactive voice response applications [VoiceXML 1.0], [VoiceXML 2.0], [VoiceXML 2.1].
But VoiceXML applications are described by context-free grammars. Thus, recognized vocabulary is limited. This paper presents an enhancement of VoiceXML. Our idea is to overlap this limitation of VoiceXML by using asynchronously capabilities of speech-dictation systems.

1.1 VoiceXML grammar

A VoiceXML grammar identifies different words or phrases that a user might say. That means that the recognized vocabulary and syntax are fixed ab initio by the programmer.

A vocal grammar is, in a way, the core of a VoiceXML application, since it determines what the user connected to the server is going to be able to say (or not). The Figure 1 presents an example of VoiceXML code in which an external grammar is invoked (see Figure 2).

```
<?xml version="1.0" encoding="iso-8859-1"?>
<vxml version="1.0">
<form>
       <field name="choice">
              <prompt>Tell me your name, please.</prompt>
              <grammar type="application/x-jsgf" src="users_names.gram"/>
                     <filled>Your name is <value expr="choice"/>.
                            <clear/>
                     </filled>
              </field>
       </form>
</vxml>
```

Figure 1: A VoiceXML code example

A vocal grammar could be given in the body of the VoiceXML script (inline grammar), or in separate file (external grammar). The Figure 2 shows the external vocal grammar used in our example. The two firsts rules are private, while the last one is public.

The sign " | " symbolizes an alternative (a logical "or"). The sign" * ", named Kleene star, indicates that the element before this symbol can appear zero or several times. With the hooks it is possible to declare an element as optional. The parentheses define a regrouping. Finally, the sign" + "indicates that element before this symbol can appear one or several times.

```
#JSGF V1.0;
grammar the_names;
<connector> = and|or|but not;
<names>=john|paul|georges|ringo;
public <sentence> = (<names> [<connector>*])+;
```

Figure 2: A Java Speech Grammar Format (JSGF) code example

As we can see on Figure 3, with the sample of dialogue managed by the vocal server, to the question pronounced by the machine "Tell me your name, please", the user can only answer a sentence satisfying the definite vocal grammar.

```
Computer: Tell me your name, please.
Human: ringo
C: Your name is ringo.

C: Tell me your name, please.
H: georges and ringo
C: Your name is georges and ringo.

C: Tell me your name, please.
H: john but not paul
C: Your name is john but not paul.

C: Tell me your name, please.
H: paul but not georges paul
C: Your name is paul but not georges paul.

C: Tell me your name, please.
H: ringo and georges but not paul and john
C: Your name is ringo and georges but not paul and john.
```

Figure 3: Example of human-machine dialogue obtained with a VoiceXML application

Grammars can range from a simple list of possible words to a complex set of possible sentences. Although VoiceXML seems to be a first interesting step towards the standardization of vocal applications on Internet, certain technical elements limit the use of VoiceXML in a natural way. The principal cause of this phenomenon is the fact that this language is mainly based on vocal technologies which can be error sourced.

1.2 Identified limits of the VoiceXML language

Hence, the performances of voice synthesis can appear disappointing when the system must face ambiguities, when it meets unknown words or proper names. The voice recognition employed is supposed being multi speaker, without enrolment phase, and being able to be realized in noisy environment. Lastly, vocal grammars are inevitably limited: the speaker will not be able to say all he/she wants, but only what will have been envisaged in design phase. This last point is a significant limit on which efforts must be dedicated.

2 Enhancement of VoiceXML

Since the first version of VoiceXML, the concept of transcription from audio to written text was evoked. Indeed, the standard proposed in appendix a <transcribe> element to add into the following version of VoiceXML. However, none the successive versions of the VoiceXML language support this tag. Moreover, <transcribe> element is not implemented by any manufacturer, and will not be available either in the future version of VoiceXML 3.0.

In order to override the limits listed above of VoiceXML, we propose to implement the <transcribe> tag. This tag could be very useful in some situations, for example to allow to the system to express some utterances not modelled by the application but pronounced by users. Then, we can provide feedback to users by re-using their input.

A possible solution consists in employing a large vocabulary voice recognition system (such as Dragon Naturally Speaking, Via Voice or Nuance), independent of that used by the VoiceXML platform, which allows the transcription of text not awaited by the grammar used in the vocal application. This speech system is used on demand by the VoiceXML Server.

2.1 Architecture

The decomposition of the proposed architecture (see Figure 4) can be seen as the following:
(1) The user access the vocal server which activates a VoiceXML application that proposes to record a free sentence for which no grammar is defined. This vocal recording is converted into an audio file;
(2) It is transmitted to a traditional server via HTTP;
(3) This audio file is then treated by an independent speech recognition tool;
(4) The obtained transcription is transmitted to the vocal server;
(5) The VoiceXML application can use the textual version of the user input.

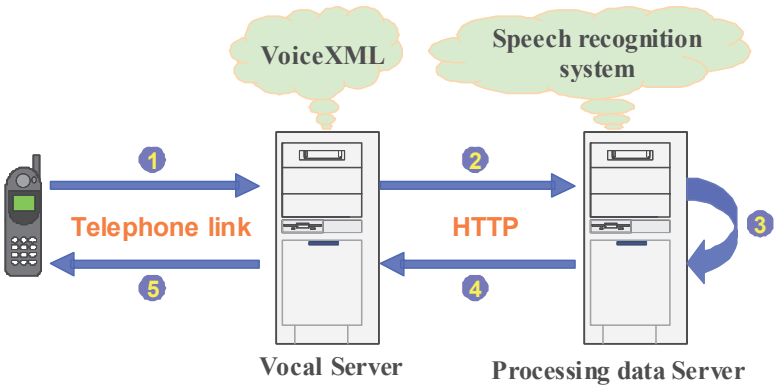

Figure 4 : Architecture for an enhancement of VoiceXML

Commercial systems usually used [Dragon], [Via Voice] based on Hidden Markov Models incorporate an acoustic model, a large vocabulary, and a linguistic model. The version of the software [Dragon] which we used for the test phase is equipped with a vocabulary gathering more than 250 000 current and specialized words.

2.2 Technical feasibility

In order to prove the technical feasibility of our project, we implemented two scenarios, one synchronous and the other, asynchronous. We used SOAP [SOAP] (and Java for the servlets) in order to develop this application as a Web service.

2.2.1 Asynchronous solution

With this approach, the treatment of the signal is differed in time. The user can freely record a sound (wav format), and submit it, via a Web page, to our service, which returns in a few seconds a written transcription of the obtained result.

2.2.2 Synchronous solution

In synchronous mode, we coupled a traditional VoiceXML application to our Web service of transcription. The vocal application allows recording a free message from any telephone. This message, converted into audio file, is sent to the Web service, in part attached SOAP. As feedback, the transcription obtained is synthesized and presented in oral form to the user. We also tested possible sequences with other Web services of translation (french/english for example) or of dictionary, which run correctly.

3 Conclusion

We explained that a strong constraint curbs the employment of VoiceXML to the use of a predetermined grammar. By coupling a traditional VoiceXML platform with an independent system of voice recognition, we showed that it is possible to increase its capacities of understanding, since a user can pronounce a sentence initially not envisaged.

Ideally, the use of a voice recognition system really independent of the speaker like the CMU Janus system [Levin et al. 2000] should improve the robustness of the whole application. A version entirely implemented with the .NET framework is in progress, and will be able to increase speed and deployment of the Web services by using WSDL [WSDL] for its description and UDDI [UDDI] for its referencing in the directories of Web services.

References

[Dragon] Dragon NaturallySpeakingTM Preferred, http://www.scansoft.com/naturallyspeaking

[Dettmer 2003] Dettmer, R., It's good to talk, speech technology for on-line services access, IEE Review, Volume: 49, Issue: 6 , Juin 2003, pp 30-33

[JSGF] JSGF: Java Speech Grammar Format, http://www.w3.org/TR/jsgf

[Levin et al. 2000] Levin L. , Lavie A. , Woszczyna M. , Gates D. , Ga-valda M. , Koll D., Waibel A., The JANUS-III translation system: Speech-to-speech translation in multipledomains, Machine translation, 2000 , vol. 15, no 1-2 , pp. 3 – 25

[Nuance GSL] GSL: Nuance Grammar Specification Language, http://studio.tellme.com/grammars/gsl

[SALT] SALT, http://www.saltforum.org

[SOAP] SOAP: Simple Object Access Protocol, http://www.w3.org/TR/soap

[UDDI] UDDI, Universal Description, Discovery and Inte-gration protocol, http://www.uddi.org

[Via Voice] IBM Via Voice, http://www.scansoft.com/viavoice

[VoiceXML 1.0] VoiceXML 1.0., W3C Recommendation, http://www.w3.org/TR/voicexml10

[VoiceXML 2.0] VoiceXML 2.0., W3C Recommendation, http://www.w3.org/TR/voicexml20

[VoiceXML 2.1] VoiceXML 2.1, Working Draft, http://www.w3c.org/TR/2004/WD-voicexml21-20040323

[WSDL] WSDL : Web Services Description Language, http://www.w3.org/TR/wsdl

[X+V] X+V, XHTML + Voice Profile, http://www.voicexml.org/specs/multimodal/x+v/12